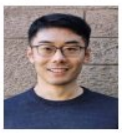
CrossCode: Multi-level Visualization of Program Execution

Authors:

Devamardeep Hayatpur

Daniel Wigdor

Haijun Xia.



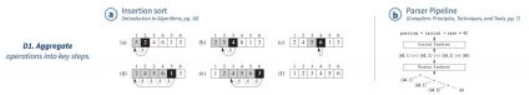


Figure shows aggregation

Figure shows abbreviation

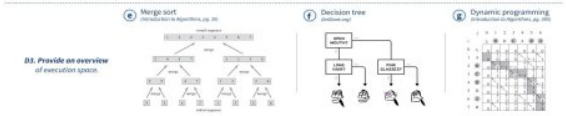


Figure shows execution overview

Results

- CrossCode offered significant advantages over traditional line-by-line debugging tools.
- Improved navigation of complex control flows and repetitive structures like loops and recursion.
- Program Understanding: CrossCode provided intuitive overviews and aggregated operations, helping participants maintain context, especially in complex code.
- Debugging:Aggregations and abbreviations streamlined debugging by highlighting key steps and skipping repetitive operations; color-coded animations aided variable tracking.
- Efficiency: CrossCode was more efficient than Python Tutor for loops and recursion, allowing users to zoom between abstraction levels and save time.

UWI
ST. AUGUSTINE CAMPUS
TRINIDAD & TOBAGO, WEST INDIES

Methodology

Research Methods

- Conducted formative studies and qualitative evaluations to design and validate CrossCode.
- Focused on how programmers reason across multiple abstraction levels.
- Aimed to enhance this reasoning process with visual tools.

Formative Study

- Analyzed 92 program visualizations, diagrams, and animations from textbooks and online tutorials.
- Identified common design patterns used to explain program behavior.
- Derived three key design patterns: aggregation, abbreviation, and execution overview.
- These patterns influenced the design of CrossCode.

Experimental Design

- Developed a custom JavaScript interpreter in TypeScript for CrossCode.
- Enabled visualization of program execution at various detail levels.
- Used control flow and data flow to help users navigate between abstractions.

Discussion

- Contributes to HCI by addressing cognitive load in navigating different levels of program execution.
- CrossCode's multi-level visualization aligns with cognitive models of how programmers reason about code.
- Offers a more flexible and intuitive interface for debugging.

Limitations

- The Learning Curve in using the program
- Data synchronization between control flow view and data view
- Limited to only Javascript

Conclusion

- CrossCode introduces a new approach to program visualization, enabling switching between abstraction levels
- Reduces cognitive load and improves debugging efficiency by aggregating and simplifying operations.
- Aligns with how programmers naturally debug and understand code, making it useful for both debugging and teaching.
- Relevant to HCI, particularly in programming environments and educational technology.

Its flexible, multi-level view could inspire innovations for novices and experts, with potential to scale to complex programs and other paradigms

LINGUISTIC DEAD-ENDS AND ALPHABET SOUP: FINDING DARK PATTERNS IN JAPANESE APPS

AUTHORS
Shun Hidaka, Sota Kobuki, Mizuki Watanabe, and Katie Seaborn,

AFFILIATIONS
Tokyo Institute of Technolgy

BACKGROUND

This paper was written by Shun Hidaka, Sota Kobuki, Mizuki Watanabe, and Katie Seaborn, all of whom are members of the Department of Industrial Engineering and Economics at the Tokyo Institute of Technology. While the first three researchers are relatively new to the field, each having authored two publications, Katie Seaborn has a more extensive body of work, with numerous publications, particularly in CHI conferences, where her research primarily focuses on human-computer interaction.

OBJECTIVE

Investigate the prevalence of dark patterns in Japanese apps and compare the similarities and differences to those in Western apps.

METHODOLOGY

In this paper the following three steps were done:

- A selection of 200 apps was made based on their ranking in the Google Play store.
- Each app was recorded for 10 minutes, during which the following six operations were performed in order:
  - Create an account and logged out
  - Close and reopen the application
  - Visit the product list page
  - Visit the settings page
  - Select name of product
  - Use the app according to its intended usage
- Pairs of three researchers then watched each recordings, identified dark patterns, and determined the specific design of each dark pattern.

RESULTS

Significantly less representation of dark patterns in the Japanese context, specifically an average of 3.9 compared to 7.4. Futher break down of results is shown below

ANALYSIS OF RESULTS

In figure 1, the bar chart shows the ratio of apperrence of each type of dark pattern, it is important to note that an ANOVA test and Shapiro-Wilk test showed no significant difference in rate of appearance for each category when compared to the other categories

In figure 2, the bar chart shows the differences in the distribution of dark patterns in Japanese apps when compared to Di Geronimo's study, a positive rate means dark patterns occurred more frequently in the Japanese apps while negative means it occured less

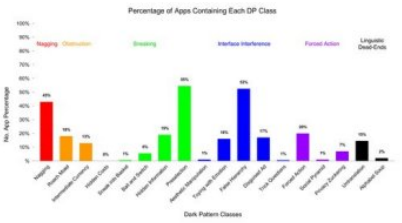


Figure 1

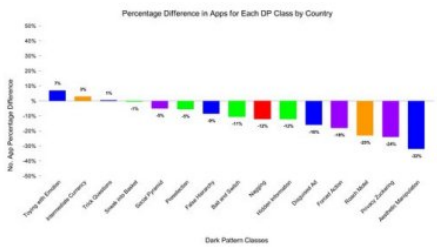


Figure 2

DISCUSSION

This paper highlights the need for more cross-cultural research on dark patterns, as it demonstrates that those found in apps from Western countries are not fully representative of all types of dark patterns. The paper also suggests that Japanese apps tend to have fewer dark patterns compared to Western apps. However, this may be because dark patterns are relatively new in Japan, implying that their prevalence could increase in the future.

CONCLUSION

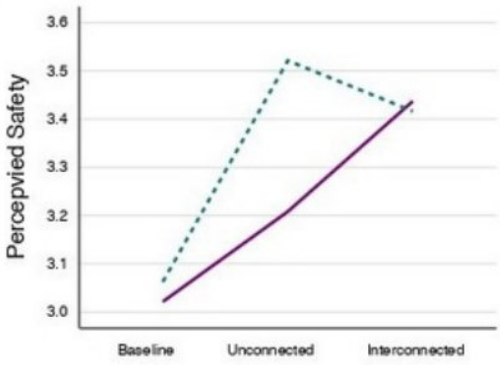
Dark Patterns appears to be more frequent in western apps and Cross-cultural research on dark patterns is need. The paper also suggests that there may be other hidden dark pattern categories within different cultural and linguistic contexts. It also shows that current research is biased toward Western countries, and there is a need for more diverse studies, as Western populations are not representative of the global population.

UWI
ST. AUGUSTINE CAMPUS

Exploring the Impact of Interconnected External Interfaces in Autonomous Vehicles on Pedestrian Safety and Experience

BACKGROUND

The authors of this paper are leading researchers in Human-Computer Interaction (HCI), including Tram Thi Minh Tran, Callum Parker, Marius Hoggenmüller, and Yiyuan Wang from the University of Sydney's Design Lab, and Martin Tomitsch from the University of Technology Sydney. Their research focuses on the impact of interconnected external Human-Machine Interfaces (eHMIs) on pedestrian safety and experience. This CHI 24' explores how these interfaces can improve communication between autonomous vehicles and pedestrians, enhancing safety in complex traffic environments.

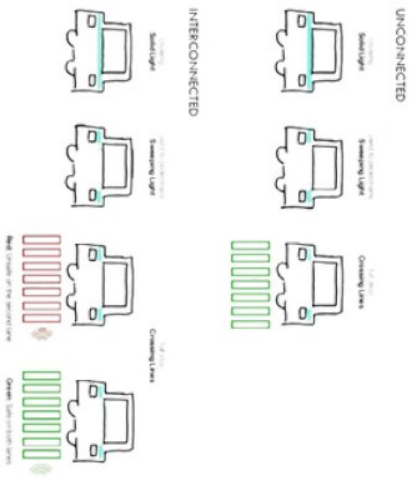


BACKGROUND

The study used a VR-based experiment with 32 participants to evaluate pedestrian safety and trust in autonomous vehicle (AV) systems. Participants experienced three conditions: no eHMI, unconnected eHMI, and interconnected eHMI. Surveys and interviews were conducted to collect both qualitative and quantitative data on their perceptions of safety and trust in these systems.

RESULTS

The study found that interconnected eHMIs improved pedestrian safety and encouraged cautious behavior. However, the use of red signals caused confusion, and the complexity of the system didn't consistently increase trust. While eHMIs reduced cognitive load in some cases, trust levels varied. Participants with prior knowledge of eHMIs had a more positive user experience.



DISCUSSION

The findings show that interconnected eHMIs improve pedestrian safety and interactions with autonomous vehicles by reducing collisions and enhancing perceived safety. This aids in building user trust and improving HCI designs for autonomous driving. However, the study's limitations include the use of VR simulations, which may not fully represent real-world behaviors, and a small sample size.

CONCLUSION

This research demonstrates the importance of designing effective eHMIs to improve pedestrian safety when interacting with AVs. Interconnected eHMIs show promise in reducing confusion in multi-lane traffic environments, but further refinements are needed.